

Simulation's Users Guide and nomenclature

1	Content of the folder "Lionel"	1
1.1	Paper.doc and Figures.doc.....	1
1.2	Simulation.ppt.....	1
1.3	Elsevier.dot.....	1
1.4	Sellmeier.pdf.....	1
1.5	Index.xls.....	1
1.6	RMS.xls.....	1
1.7	Results.zip.....	1
1.8	Results EF.zip.....	2
1.9	R3809U-50_TPMH1067E08.pdf.....	2
1.10	Folders Simulation/ and Simul2e-/.....	2
2	Simulation's Users Guide.....	3
2.1	Description of the source code files.....	3
2.1.1	<i>main.cpp</i>	3
2.1.2	<i>parameters.h</i>	3
2.1.3	<i>PC-MCP.cpp/h</i>	3
2.1.4	<i>window-PC.cpp/h</i>	3
2.1.5	<i>transSilica.dat and transMgF2.dat</i>	3
2.1.6	<i>R3809U-58.dat</i>	3
2.2	Personalizing the parameters.....	3
2.2.1	<i>Gap size</i>	3
2.2.2	<i>Electric field</i>	3
2.2.3	<i>Changing the magnitude/angular distribution</i>	3
2.2.4	<i>Changing a parameter</i>	4
2.2.5	<i>Number of simulated electrons and precision</i>	4
2.3	Execution of the program.....	4
2.4	Data files.....	5
2.5	Deeper in the program.....	5
3	Simulation of 2 electrons with the Coulomb repulsion.....	6
4	Beam incident angle discussion.....	6

1 Content of the folder “Lionel”

All the files I left are in the folder called “*Lionel*” on the Desktop when Jean-François Genat’s session is opened on the computer at the left of the printer called “EDGE 4”.

1.1 *Paper.doc and Figures.doc*

My publication (**to be updated**) and the figures I planned to include in it.

1.2 *Simulation.ppt*

PowerPoint of my presentation (**to be updated**).

1.3 *Elsevier.dot*

Model I used in my publication.

1.4 *Sellmeier.pdf*

PDF document in which could be found the Sellmeier dispersion equation constants for the fused silica and the magnesium fluoride (MgF_2). This helps me calculate the optical indexes in *Index.xls* (see paragraph 1.5) and so as to update the file *R3809U-58.dat* (see chapter 2.1.6).

1.5 *Index.xls*

Excel file which contains the plot of the optical indexes of the fused silica and the MgF_2 (ordinary and extraordinary indexes) in the range $[0.2 \mu\text{m}, 7 \mu\text{m}]$.

‘Simulation’ indicates the original data used in the ROOT simulation of the window, the others (O, E and ‘Internet’) the ones I calculated (cf. *Sellmeier.pdf* in paragraph 1.4).

The limit $\sqrt{2}$, above which no Čerenkov Effect is possible, is represented, too.

1.6 *RMS.xls*

Excel file containing all the RMS (in time, position and angle) values given by my simulation vs. the gap size for the different angular distributions (with $E = 100 \text{ V/mm}$, data files stored in *Results.zip*) and then for different values of the electric field (arcsine distribution only, data files stored in *Results EF.zip*).

1.7 *Results.zip*

Result files for $E = 100 \text{ V/mm}$ and the 5 angular distributions. For a description of the content of the data files, please refer to the chapter 2.4. Nomenclature: **D- α -g.txt**

➤ **D:** distribution used:

- ✓ f: flat
- ✓ a: arcsine
- ✓ 2: Gaussian (1%)

- ✓ 3: Gaussian (1‰)
- ✓ 100: pseudo-Dirac
- **α**: physical quantity
 - ✓ angle
 - ✓ time
 - ✓ x: position
- **g**: gap size

1.8 Results EF.zip

Result files for the arcsine distribution and different electric field values. For a description of the content of the data files, please refer to the chapter 2.4. Nomenclature: **E-α-g.txt**

- **E**: electric field value in V/mm
- **α**: physical quantity
 - ✓ angle
 - ✓ time
 - ✓ x: position
- **g**: gap size

1.9 R3809U-50_TPMH1067E08.pdf

PDF document from Hamamatsu where the Figure 1 (photocathode radiant sensitivity vs. wavelength) helped me to recalculate the quantum efficiency of an MgF₂ window. The result is the file *R3809U-58.dat* in the folder *Simulation/* (a copy of this file could be found in the folder *Simul2e-/*).

From a radiant sensitivity $S(\lambda)$, the quantum efficiency QE is calculated with the formula:

$$QE = \frac{S(\lambda) \times 1.24}{\lambda}$$

with QE in the range [0,1], $S(\lambda)$ in mA/W and λ in nm.

1.10 Folders Simulation/ and Simul2e-/

The content of these folders is described in the chapters 2.1 and 3.

2 Simulation's Users Guide

2.1 Description of the source code files

These C++ files are in the folder called *Simulation/*. Here is a short description of them.

2.1.1 *main.cpp*

This file contains the *main* function. It calls all the other functions to calculate the final state of each electron, their time of flight, the RMS and create & fill the data files.

2.1.2 *parameters.h*

This file contains all the physical constants such as c_0 , β , π and h , some calculations constants and the parameters like the electric field or the size of the gap.

2.1.3 *PC-MCP.cpp/.h*

All the functions used to calculate what happens (final state, ToF, RMS) between the photocathode and the first MCP's surface are in this file.

2.1.4 *window-PC.cpp/.h*

I wrote back some functions used in the ROOT simulation of the Čerenkov Effect, of the window and the photocathode in C++, so as to integrate them in my simulation. In the calculation of the optical indexes, I updated the Sellmeyer dispersion equation constants. To be continued...

"*window-PC.h*" is already included in the file *main.cpp*, even if none of its function is used.

2.1.5 *transSilica.dat and transMgF2.dat*

These files are identical to the ones coming from the ROOT simulation. They contain the transmittance vs. the wavelength of the two kinds of window: fused silica and magnesium fluoride.

2.1.6 *R3809U-58.dat*

This file has the same origin as the 2 previous ones. I updated this one, which contains the quantum efficiency vs. the wavelength. See chapter 1.9 for more details.

2.2 Personalizing the parameters

2.2.1 *Gap size*

Like most of the parameters, it is located in the file *parameters.h*, line 24:

```
#define GAP .002
```

ATTENTION: the gap size should be given in meters (2 mm here)!

2.2.2 *Electric field*

It is located just under the gap size, in *parameters.h*, line 25:

```
#define EF 100000
```

ATTENTION: the electric field here is in V/m (100 V/mm here).

2.2.3 *Changing the magnitude/angular distribution*

This could be found in the file *main.cpp*, lines 64 and 65 respectively.

```
electron.v_r0 = V0_RANDOM * rand();  
electron.v_theta0 = asin(RAND2*rand() - 1);
```

To choose one of the distributions I used, the second line should be replaced by:

- flat : electron.v_theta0 = PI/2.*(RAND2*rand() - 1);
- arcsine : electron.v_theta0 = asin(RAND2*rand() - 1);
- Gaussian (1%) : electron.v_theta0 = PI/2. * randGauss(2.576);
- Gaussian (1‰) : electron.v_theta0 = PI/2. * randGauss(); OR randGauss(3.290);
- Dirac : electron.v_theta0 = PI/2. * randGauss(100.);

2.2.4 *Changing a parameter*

To optimize the precision of calculation and of the histogram, for each physical quantity, a minimum and maximum should be defined. And of course each time a parameter is changed, these limits change, too.

They are in the file *parameters.h* lines 40 to 50:

```
#define T_MIN (4.57e-10)  
#define T_MAX (4.78e-10)  
#define T_sym false  
  
#define X_MIN 0.  
#define X_MAX (1.66e-4)  
#define X_sym true  
  
#define A_MIN 0.  
#define A_MAX 0.0415  
#define A_sym true
```

T_MIN and T_MAX are in seconds, X_MIN and X_MAX in meters, and A_MIN and A_MAX in radians. Since the problem is symmetric in position and in angle (wrt 0), X_MIN and A_MIN should have a positive value.

2.2.5 *Number of simulated electrons and precision*

These parameters does not affect the physics of the problem and therefore do not imply to change the limits (see paragraph 2.2.4).

The number of simulated electron, or number of event is called NEVT and could be changed in *parameters.h*, line 36: #define NEVT 1000000000

The program has been written so as its maximum value is 18,446,744,073,709,551,615 which is a limit in C++; to use such a number, the variable type should be *unsigned long long int*.

The range of study (see paragraph 2.2.4) is divided in PRECISION sub-ranges for the histogram and to calculate the RMS. This parameter is two lines below NEVT:

```
#define PRECISION 5000
```

2.3 Execution of the program

When executing the program, several informations are displayed; here is a description of the different parts:

➤ the green part means that most of the variables have

```
C:\Documents and Settings\genat\Bureau\Lionel\Simulation>Simulation.exe
===== Photocathode gap simulation =====
```

```
Initialization... done
```

```
Beginning of calculation... Please wait
0% done
```

```
===== METADATA: 'angle' histogram
=====
==
== A_MIN (in radians):      0
== A_MAX (in radians):    0.0415
== da (in radians):       8.3e-006
== PRECISION_A:          5000
== Nb of simulated electrons: 1000000000
== Gap size (in meters):  0.002
== Nb electrons out of range: 0
== RMS (in radians):      0.0137838028582222
== Medium angle (in radians): 0
==
=====
=====
0      1923683
1      1631402
2      1524613
3      1457569
```

```
===== METADATA: 'X' histogram
=====
==
== X_MIN (in meters):      0
== X_MAX (in meters):    0.000166
== dX (in meters):       3.32e-008
== PRECISION_X:          5000
== Nb of simulated electrons: 1000000000
== Gap size (in meters):  0.002
== Nb electrons out of range: 0
== RMS (in meters):      5.41628395183137e-005
== Medium 'X' (in meters): 0
==
=====
=====
0      1932968
1      1637986
2      1533766
3      1465782
```

```
===== METADATA: 'time' histogram
=====
==
== T_MIN (in seconds):    4.57e-010
== T_MAX (in seconds):    4.78e-010
== dt (in seconds):       4.2e-015
== PRECISION_T:          5000
== Nb of simulated electrons: 1000000000
== Gap size (in meters):  0.002
== Nb electrons out of range: 0
== RMS (in seconds):      5.05231829085597e-012
== Medium time (in seconds): 4.69231234934873e-010
==
=====
=====
0      0
1      0
2      0
3      0
```

been initialized and that the simulation began;

➤ the yellow part shows the progression of the simulation;

➤ then the calculation of the 3 RMS begins and then the 3 data files are created;

➤ this gives the number of events out of the range (see paragraph 2.2.4).

2.4 Data files

(Almost) all the useful informations concerning a run are given in the header of each data file. When I began to change the electric field, I noticed that it was not given in the header: thus I made the change although the headers of all my data files (in *Results.zip* and *Results EF.zip*) do not have it.

Here are 3 examples of header:

$$d\alpha = \frac{\alpha_{MAX} - \alpha_{MIN}}{PRECISION_{\alpha}}, \alpha \equiv A, X \text{ or } T.$$

The name PRECISION_A/X/T is a remnant from the former definition of PRECISION (see paragraph 2.2.5), but they all mean the same.

From the 14th line (15th when the information about the electric field is given) begin the real data: the first column gives the index and the second the value (see paragraph 2.5 for extra details).

From the index, the original value could be found back with multiplying i by $d\alpha$, and then by adding to the result α_{MIN} .

All the data files I obtained are stored in 2 archives:

- *Results.zip*: $E = 100$ V/mm and the 5 distributions are used;
 - *Results EF.zip*: arcsine distribution only and the electric field varies.
- For the nomenclature used in these archives, please refer to the chapters 1.7 and 1.8.

2.5 Deeper in the program

For each of the NEVT simulated electrons, the initial conditions are set: position (0,0) and velocity as shown in the paragraph 2.2.3. The function fTOF is then called: for a given electron (a structure, given in argument), it calculates the final state (i.e. `electron.x`, `electron.z`, `electron.v_r` and `electron.v_theta`) and return the time of flight of this electron. The tables `nTOF[]`, `nX[]` and `nA[]` are then incremented.

The index (i) of the 3 tables `nTOF[]`, `nX[]` and `nA[]` refers to a value (respectively a ToF, a position and an angle) and the value of the table (respectively `nTOF[i]`, `nX[i]` and `nA[i]`) gives the number of times the value referred by i is obtained.

Thus for the calculated ToF, the corresponding value of i is calculated ((int) ((time_TOF - T_MIN)/dt)). If the calculated ToF is out of range, the variable `n_T` is incremented; else `nTOF[i]` is incremented. . Then the same process start back with the final position (, `nX[i]` and `n_X`) and the final angle (, `nA[i]` and `n_A`).

The last thing, once the NEVT electrons have been simulated, is to calculate from these tables the RMS through the function called RMS.

In the data files, the “real data” are in fact the list

$$i \quad n\alpha[i]$$

for i from 0 to PRECISION-1.

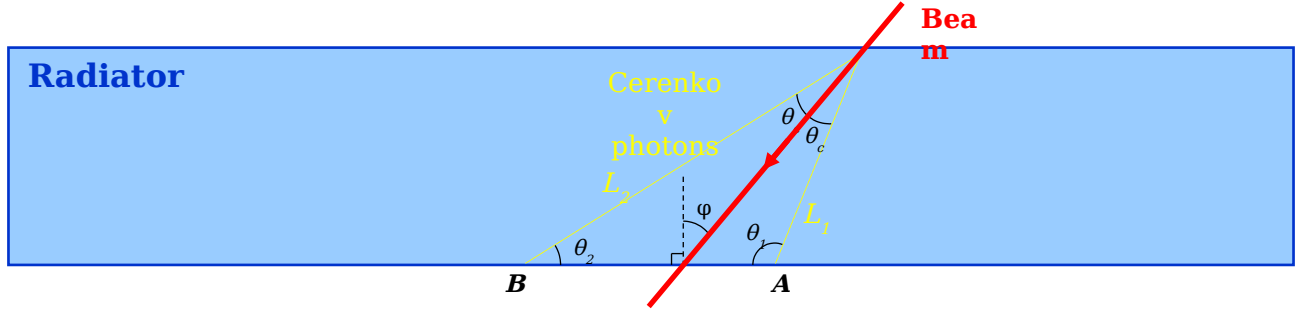
3 Simulation of 2 electrons with the Coulomb repulsion

In the folder *Simul2e-/* could be found the same files than in the folder *Simulation/* (the *.dat* files are identical, but the code files (*.cpp* and *.h*) contain a different code). 2 new files could be found: *calculation.cpp* and *calculation.h*: I just planned to put in it the functions needed to make “basic” calculation like the distance between the 2 electrons or the vector joining one electron with the other...

This work is far from the end...

4 Beam incident angle discussion

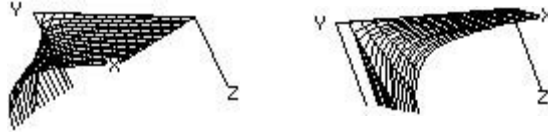
Here, φ could be considered as the error on the “normal incidence” of the beam wrt the radiator.



AN:

$n = \sqrt{2}$, $\beta = 1$, $d = 10$ mm (almost the worst configuration)

φ	Δt
1°	2.33 ps
$(\pi/180)$	
0.43°	1 ps
0.043	0.1 ps



X: gap size; Y: φ ; Z: Δt .

These times should be added to the TTS obtained with my simulation.

$$\theta_c = \arccos\left(\frac{1}{\beta n}\right)$$

$$\begin{cases} \theta_1 = \frac{\pi}{2} + \varphi - \theta_c \\ \theta_2 = \frac{\pi}{2} - \varphi - \theta_c \end{cases} \Rightarrow \begin{cases} L_1 = \frac{d}{\cos(\varphi - \theta_c)} \\ L_2 = \frac{d}{\cos(\varphi + \theta_c)} \end{cases} \Rightarrow \begin{cases} t_1 = \frac{nL_1}{c_0} \\ t_2 = \frac{nL_2}{c_0} \end{cases} \Rightarrow \Delta t = \frac{nd}{c_0} \left(\frac{1}{\cos(\varphi - \theta_c)} - \frac{1}{\cos(\varphi + \theta_c)} \right)$$